Delay-Insensitive Pipelined Communication on Parallel Buses

Mario Blaum, *Senior Member, IEEE*, and Jehoshua Bruck, *Senior Member, IEEE*

***Abstract*—Consider a communication channel that consists of several subchannels transmitting simultaneously and asynchronously. As an example of this scheme, we can consider a board with several chips. The subchannels represent wires connecting between the chips where differences in the lengths of the wires might result in asynchronous reception. In current technology, the receiver acknowledges reception of the message before the transmitter sends the following message. Namely, pipelined utilization of the channel is not possible.**

Our main contribution is a scheme that enables transmission without an acknowledgment of the message, therefore enabling pipelined communication and providing a higher bandwidth. Moreover, our scheme allows for a certain number of transitions from a second message to arrive before reception of the current message has been completed, a condition that we call skew. We have derived necessary and sufficient conditions for codes that can tolerate a certain amount of skew among adjacent messages (therefore, allowing for continuous operation) and detect a larger amount of skew when the original skew is exceeded. These results generalize previously known results.

We have constructed codes that satisfy the necessary and sufficient conditions, studied their optimality, and devised efficient decoding algorithms. To the best of our knowledge, this is the first known scheme that permits efficient asynchronous communications without acknowledgment. Potential applications are in on-chip, on-board, and board to board communications, enabling much higher communication bandwidth.

***Index Terms*—Parallel communication, skew, pipelined channel, error-correcting codes, asynchronous communication.**

I. Introduction

A. Motivation and Background

CONSIDER a communication channel that consists of several subchannels transmitting simultaneously. As an example of this scheme consider a board with several chips where the subchannels represent wires connecting between the chips and differences in the lengths of the wires might result in asynchronous reception. Namely, we would like to transmit a binary vector of length n using n parallel channels/wires. Every wire can carry only one bit of information. Each wire represents a coordinate of the vector to be transmitted. In this model, an electrical transition corresponds to a 1, while absence of a transition corresponds to a 0. The propagation delay in the wires varies. The problem is to find an efficient communication scheme that will be delay-insensitive.

Clearly, this problem is very common and arises in every system that incorporates transmission of information over parallel lines. Currently, there are two approaches for solving it in practice:

1) There is a clock that is shared by both the transmitter and the receiver, and the state of the wire at the time of the clock represents the corresponding bit of information. This is a synchronous type of communication (which is not always feasible due to the difficulties in clock distribution and the fact that the transmitter might be part of an asynchronous system).
2) Asynchronous type of communications. Here the idea is to send one vector at a time and have a handshake mechanism. Namely, the transmitter sends the following vector only after getting an acknowledgment that the current vector was completely received by the receiver.

A natural question with regard to the asynchronous type of communication is: How does the receiver know that the reception is complete? This problem was studied by Verhoeff [9]. He describes the foregoing physical model as a scheme in which the sender communicates with the receiver via parallel tracks by rolling marbles (that correspond to a logical 1) in the tracks. The assumption of rolling marbles is equivalent to the transmission of electrical transitions. Although the marbles are sent in parallel, the channels are asynchronous. This means that marbles are received randomly and at different instants.

Before presenting Verhoeff's result we introduce some notation. Let us represent the channels with the numbers $1, 2, \ldots, n$. After the mth transition has arrived, the receiver obtains a sequence $\hat{X}_m = x_1, x_2, \ldots, x_m$, where $1 \le x_i \le n$, and x_i represents the fact that the ith transition was received at the x_ith channel. The set $\{x_1, x_2, \ldots, x_m\}$ is the support (i.e., the set of nonzero coordinates) of a vector, and it determines uniquely a binary vector. From now on, $\hat{X}_m = x_1, x_2, \ldots, x_m$ denotes a sequence as defined above, and $X_m = \{x_1, x_2, \ldots, x_m\}$ the binary vector as defined by its support corresponding to sequence $\hat{X}_m$. For instance, assume that we have five channels and we receive the sequence $\hat{X}_4 = 2, 3, 2, 4$. This means the first transition arrived in channel 2, the second one in channel 3, the third one in channel 2, and the fourth one in channel 4. The support of the corresponding binary vector is $X_4 = \{2, 3, 4\}$ (repeated arrivals count only once!), and the binary vector

Manuscript received Apr. 8, 1993.

M. Blaum is with IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, CA 95120; e-mail blaum@almaden.ibm.com.

J. Bruck is with the California Institute of Technology, MC 116-81, Pasadena, CA 91125; e-mail bruck@systems.caltech.edu.

IEEECS Log Number C95030.

itself is $X_4 = 0\ 1\ 1\ 1\ 0$. In words, capital letters with a hat will denote sequences, while capital letters denote either vectors or their supports.

The following example shows the difficulty of choosing indiscriminate vectors for parallel asynchronous communications. Assume that a vector $X = 0110$ and a vector $Y = 0100$ are transmitted in some order. In the language of sets we have $X = \{2, 3\}$ and $Y = \{2\}$. When the receiver gets a transition in channel number 2, it is not clear whether it just received Y or it should wait to get a transition in channel 3 (this will correspond to receiving X).

In general, the parallel asynchronous transmission model considered in [9], is the following: Assuming that a vector X is transmitted, once reception has been completed, the receiver acknowledges receipt of the message. The next message is sent by the sender only after the receipt of the acknowledgment. The problem is finding a code C whose elements are messages such that the receiver can identify when transmission has been completed. It is easy to see, as shown in [9] and as suggested in the example above, that the codes having the right property are the so-called *unordered codes*, i.e., all its elements are unordered vectors (we say that two binary vectors are unordered when their supports are unordered as sets—one set is not a subset of the other).

One of the disadvantages of using the asynchronous type of communication is the fact that the channel is not fully utilized. Namely, there is at most one vector in the wires at any given time. This becomes very critical when the transmission rates are getting higher and lines are getting longer.

B. The New Paradigm

In this paper, we present a novel scheme that enables a pipelined utilization of the channel. In addition, our scheme has the important feature of not using a handshake (acknowledgment) mechanism. Hence, there is no need in communication between receiver and sender.

We note here that if one is ready to pay in performance, then a possible strategy, if acknowledgment of messages is not allowed, is that the sender will wait long enough between messages. So, if the sender sends a codeword X followed by a codeword Y, it will be very unlikely that a transition from Y will arrive before the reception of X has been completed. With this scheme, we can again use unordered codes as in [9].

The purpose of this paper is to study parallel asynchronous pipelined communication without acknowledgment. The main difficulty in this scheme is that a certain number of transitions from the second message might arrive before reception of the current message has been completed, a condition that we call *skew*.

We give next a precise mathematical definition of the concept of skew. Assume that a vector X is transmitted followed by other vectors, say Y and W. At reception, we obtain a sequence $\hat{Z} = x_1, x_2, \ldots, x_i, \ldots$. If there is no skew of X with respect to $\hat{Z}$, all the transitions from X arrive first and then the transitions from the next messages. However, this is not the case when there is skew. Throughout the paper, we will make the following assumption: The skew occurs only between adjacent vectors. For instance, a transition belonging in Y may arrive before all the transitions in X have arrived, but not a transition from W. Similarly, a transition belonging in W may arrive before all the transitions in Y have arrived, but not a transition from the vector following W.

There are two parameters that are related to the skew of X with respect to $\hat{Z}$. The first one, denoted $m(X; \hat{Z})$, represents the index of the last transition in X before the occurrence of skew, i.e., the last transition in X before the arrival of either a transition not in X (meaning a transition in $Y - X$) or a repeated arrival (which is in $Y \cap X$). The second one, denoted $r(X; \hat{Z})$, represents the index of the last arrival in X. If there is no skew, then $m(X; \hat{Z}) = r(X; \hat{Z})$. For instance, if $X = \{1, 2, 4\}$, $Y = \{1, 3, 5\}$ and $\hat{Z} = 2, 3, 1, 1, 4, 5, \ldots$, we can see that $m(X; \hat{Z}) = 1$ and $r(X; \hat{Z}) = 5$. Notice that transitions $3 \in Y - X$ and $1 \in Y \cap X$ have arrived before the completion of X, which occurs when transition 4 arrives.

More precisely, if $\hat{Z} = x_1, x_2, \ldots, x_i, \ldots$ is a sequence, $\hat{Z}_j = x_1, x_2, \ldots, x_j$, Z_j denotes the vector corresponding to $\hat{Z}_j$, $j \geq 1$, and X is a vector, then

$$m\left(X; \hat{Z}\right) = \min\left\{j : Z_j \subseteq X \text{ and } \left(x_{j+1} \notin X \text{ or } x_{j+1} \in Z_j\right)\right\} \quad (1)$$

and

$$r\left(X, \hat{Z}\right) = \min\left\{j : X \subseteq Z_j\right\} \quad (2)$$

Notice that if $x_1 \notin X$, $m(X; \hat{Z}) = 0$.

We are ready now to define the concept of skew of a vector X with respect to a sequence $\hat{Z}$.

DEFINITION 1.1. *Let X be a subset of $\{1, 2, \ldots, n\}$ (equivalently, X is a binary vector of length n). Let $\hat{Z} = x_1, x_2, \ldots, x_j, \ldots$ be a sequence whose elements are in $\{1, 2, \ldots, n\}$, $\hat{Z}_i = x_1, x_2, \ldots, x_i$ and Z_i the set corresponding to $\hat{Z}_i$. Let $m = m(X; \hat{Z})$ and $r = r(X; \hat{Z})$ be as defined by* (1) *and* (2), *respectively.*

We say that the skew of X with respect to $\hat{Z}$ is equal to (l_1, l_2) (notation, $S(X; \hat{Z}) = (l_1, l_2)$), if and only if

$$l_1 = |(Z_r - Z_m) \cap X| \quad \text{and} \quad l_2 = r - m - l_1,$$

where $|S|$ denotes the cardinality of a set S. Notice that repeated arrivals are counted towards l_2.

Let $S(X; \hat{Z}) = (l_1, l_2)$. We say that $S(X; \hat{Z})$ does not exceed (s_1, s_2), denoted $S(X; \hat{Z}) \leq (s_1, s_2)$, if $l_1 \leq s_1$ and $l_2 \leq s_2$. Otherwise, we say that $S(X; \hat{Z})$ exceeds (s_1, s_2) (notation, $S(X; \hat{Z}) > (s_1, s_2)$).

Given $S(X; \hat{Z}) = (l_1, l_2)$, the parameter l_1 measures the number of transitions missing in X when the first transition not in X arrives. The parameter l_2 measures the number of transitions not in X and repeated arrivals that arrive before reception of X has been completed.

The next example illustrates the definition of skew.

EXAMPLE 1.1. Assume that $X = 11000$ is transmitted followed by other vectors. As a set, $X = \{1, 2\}$. At reception, assume that the sequence $\hat{Z} = 231425\ldots$ is obtained. Equations (1) and (2) give $m = m(X; \hat{Z}) = 1$ and $r = r(X; \hat{Z}) = 3$, respectively. Therefore, we obtain $Z_m = Z_1 = \{2\}$ and $Z_r = Z_3 = \{1, 2, 3\}$, giving $Z_r - Z_m = Z_3 - Z_1 = \{1, 3\}$.

According to Definition 1.1, $l_1 = |(Z_r - Z_m) \cap X| = |\{1\}| = 1$ and $l_2 = r - m - l_1 = 1$, so $S(X; \hat{Z}) = (1, 1)$.

Similarly, if we receive $\hat{Z} = 224135$, we can see that $m = m(X; \hat{Z}) = 1$ and $r = r(X; \hat{Z}) = 4$. Now, we obtain $Z_m = Z_1 = \{2\}$ and $Z_r = Z_4 = \{1, 2, 4\}$, giving $Z_r - Z_m = Z_4 - Z_1 = \{1, 4\}$.

According to Definition 1.1, $l_1 = |(Z_r - Z_m) \cap X| = |\{1\}| = 1$ and $l_2 = r - m - l_1 = 2$, so $S(X; \hat{Z}) = (1, 2)$.

The next step is defining codes that can either detect or correct skew. Our approach to dealing with skew is to use coding theory methodology and identify the properties of a family of vectors (a code) that can handle the skew. We want codes that can either detect or tolerate up to a certain amount of skew, or simultaneously tolerate and detect skew (compare with codes that can simultaneously correct and detect errors). Formally:

DEFINITION 1.2. *Let t_1, t_2, s_1, s_2 be four non-negative integers and let C be a code. Let X, Y, W, ... be codewords in C, and assume that X is transmitted followed by Y and then by W which is followed by other codewords, and that no transition in W arrives before the reception of X is completed. Let $\hat{Z}$ be the received sequence. Then:*

1) *We say that C is (t_1, t_2)-*skew-detecting (SD) *if the code will correctly decode X when $S(X; \hat{Z}) = (0, 0)$ (i.e., no skew), and will detect the occurrence of skew as long as $(0, 0) < S(X; \hat{Z}) \le (t_1, t_2)$.*
2) *We say that C is (t_1, t_2)-*skew-tolerant (ST) *if the code will correctly decode X when $S(X; \hat{Z}) \le (t_1, t_2)$.*
3) *We say that C is (t_1, t_2)-*ST *$(t_1 + s_1, t_2 + s_2)$*-SD *if the code will correctly decode X when $(0, 0) \le S(X; \hat{Z}) \le (t_1, t_2)$ and will detect the occurrence of skew as long as $(t_1, t_2) < S(X; \hat{Z}) \le (t_1 + s_1, t_2 + s_2)$.*

SD and ST codes were studied in [2]. Here, we generalize these results and address the combination of correction and detection, namely, ST-SD codes. Notice that, in particular, an (s_1, s_2)-SD code is a (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD code with $t_1 = t_2 = 0$, and a (t_1, t_2)-ST code is a (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD code with $s_1 = s_2 = 0$.

Next, we illustrate Definition 1.2 with an example.

EXAMPLE 1.2. Consider the following code: $C = \{X, Y\}$ where $X = 10000$ and $Y = 01111$.

Claim: C is (1, 1)-ST (2, 2)-SD. In effect, consider the following decoding algorithm:

1) If the first or the second transition arrive in track 1, then conclude that X was transmitted first.
2) If the third transition is received in track 1, the decoder is unable to determine if X was transmitted followed by Y or conversely, so an error is detected. Notice that in this case, if we denote by $\hat{Z}$ the received sequence (i.e., $\hat{Z} = x_1, x_2, 1, x_3, x_4, \ldots, \{x_1, x_2, x_4, x_5\} = Y$), then $S(X; \hat{Z}) = (1, 2)$ and $S(Y; \hat{Z}) = (2, 1)$.
3) If the first three transitions arrive in tracks 2 to 5, then conclude that Y was transmitted first.

We can see that the decoding algorithm above will correct skew not exceeding (1, 1) and will detect skew exceeding (1, 1) but not (2, 2).

Although Example 1.2 is very simple, the reader is urged to comprehend it, since the general case involves a similar reasoning. The necessary and sufficient conditions for a code to be (t_1, t_2)-SD $(t_1 + s_1, t_2 + s_2)$-ST, to be given in the next section, will allow us to readily explain why the code in Example 1.2 is (1, 1)-ST (2, 2)-SD.

C. Contributions and Organization

Clearly, it is not enough to just define (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD codes. Our real goal is to identify the properties that characterize those codes and use them for constructions. Indeed, we were able to derive necessary and sufficient conditions for (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD codes. These conditions are given using global distance properties between codewords. They fully characterize a set of vectors that can enable operation in the desired new paradigm.

We also provide efficient encoding and decoding algorithms.

In summary, we have used coding theory methodologies in order to create an efficient scheme for delay-insensitive parallel pipelined asynchronous communication. As it turned out, new families of codes as well as new encoding and decoding algorithms are needed in order to address this problem.

In the next section, we prove the characterization theorem for (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD codes and present an algorithm for correction and detection of skew. We also study particular cases of the general characterization theorem, and we verify that they coincide with known results.

In Section III, we address the issue of actual code constructions.

II. CHARACTERIZATION THEOREM AND DECODING ALGORITHM FOR (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD CODES

In this section, we give a characterization in terms of distance between codewords of (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD codes (Definition 1.2), starting with necessary conditions and then proving that these conditions are also sufficient. The sufficient conditions are proven by providing a decoding algorithm, and showing that the decoding algorithm correctly decodes a codeword when the skew does not exceed (t_1, t_2), and detects the presence of skew when this skew exceeds (t_1, t_2) but not $(t_1 + s_1, t_2 + s_2)$.

Given two binary vectors X and Y of length n, we denote by $N(X, Y)$ the number of coordinates in which X is 1 and Y is 0

[5]. For example, if $X = 10110$ and $Y = 00101$, we have $N(X, Y) = 2$ and $N(Y, X) = 1$. Notice that $N(X, Y) + N(Y, X) = d_H(X, Y)$, where d_H denotes Hamming distance. In the language of sets, $N(X, Y) = |X - Y|$.

A. The Necessary Condition

The following theorem gives necessary conditions for a code to be (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD.

THEOREM 2.1. *Let C be a (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD code. Let $t = \min\{t_1, t_2\}$, $T = \max\{t_1, t_2\}$, $s = \min\{s_1, s_2\}$, $S = \max\{s_1, s_2\}$, $\tau = \min\{t_1 + s_1, t_2 + s_2\}$ and $\rho = \max\{t_1 + s_1, t_2 + s_2\}$. Let X and Y be arbitrary distinct codewords in C with $N(X, Y) \le N(Y, X)$, then:*

a) If $(t_1 - t_2)(s_1 - s_2) \ge 0$, then at least one of the following three conditions occurs:

1) $N(X, Y) \ge \tau + 1$.
2) $N(X, Y) \ge T + 1$ *and* $N(Y, X) \ge \rho + 1$.
3) $N(X, Y) \ge 1$ *and* $N(Y, X) \ge t_1 + t_2 + S + 1$.

b) If $(t_1 - t_2)(s_1 - s_2) < 0$, then at least one of the following four conditions occurs:

1) $N(X, Y) \ge \tau + 1$.
2) $N(X, Y) \ge T + 1$ *and* $N(Y, X) \ge \rho + 1$.
3) $N(X, Y) \ge t + 1$ *and* $N(Y, X) \ge \max\{\rho + 1, t_1 + t_2 + s + 1\}$.
4) $N(X, Y) \ge 1$ *and* $N(Y, X) \ge t_1 + t_2 + S + 1$.

PROOF. Given two codewords X and Y in the code C, let $I = X \cap Y$, $D = X - Y$ and $E = Y - X$. By $\hat{I}$, $\hat{D}$, and $\hat{E}$, we denote sequences of the elements of I, D, and E in some order.

a) Assume first that

$$(t_1 - t_2)(s_1 - s_2) \ge 0. \tag{3}$$

We will assume that the conditions are not true and show that the code is unable to correct/detect the specified skew. Assume that there exist two distinct X and Y with $N(X, Y) \le N(Y, X)$ such that:

$$N(X, Y) \le \tau, \tag{4}$$

and

$$N(X, Y) \le T \text{ or } N(Y, X) \le \rho, \tag{5}$$

and

$$N(X, Y) \le 0 \text{ or } N(Y, X) \le t_1 + t_2 + S. \tag{6}$$

Notice that $N(X, Y) = 0$ cannot occur. If that were the case, $X \subseteq Y$. Then, if a sequence $\hat{Z} = \hat{X}, \hat{E}, \hat{X}$ is received, by examining $\hat{Z}$, the decoder is unable to determine whether X or Y was transmitted first. Therefore, we can replace (6) by

$$N(Y, X) \le t_1 + t_2 + S. \tag{7}$$

Combining (4), (5), and (7), we obtain

$$N(X, Y) \le \min\{\tau, T\} \text{ and } N(Y, X) \le t_1 + t_2 + S \tag{8}$$

or

$$N(X, Y) \le \tau \text{ and } N(Y, X) \le \rho. \tag{9}$$

We will show that both (8) and (9) contradict the fact that C is (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD.

Assume first that X and Y satisfy (8). There are two cases: $S = s_1$ and $S = s_2$.

If $S = s_1$, by (3), we have $s = s_2$, $T = t_1$, $t = t_2$, $\rho = t_1 + s_1$ and $\tau = t_2 + s_2$. Therefore, (8) becomes

$$|D| = N(X, Y) \le \min\{t_2 + s_2, t_1\} \text{ and } |E| = N(Y, X) \le t_1 + t_2 + s_1. \tag{10}$$

Let $E = A \cup B$, where $A \cap B = \varnothing$, $|A| \le t_2$ and $|B| \le t_1 + s_1$. Now, assume that the following sequence is received:

$$\hat{Z} = \hat{I}, \hat{A}, \hat{D}, \hat{B}, \hat{I}, \ldots \tag{11}$$

From (11), we can see that $S(Y; \hat{Z}) = (|B|, |D|)$. Since $|B| \le t_1 + s_1$ and, by (10), $|D| = N(X, Y) \le t_2 + s_2$, we have

$$S(Y; \hat{Z}) \le (t_1 + s_1, t_2 + s_2). \tag{12}$$

Therefore, since code C is (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD, by examining $\hat{Z}$, the decoder will either decide that Y was the transmitted codeword or it will detect an error.

On the other hand, using again (10) and the fact that $|A| \le t_2$, we have

$$S(X; \hat{Z}) = (|D|, |A|) = (N(X, Y), |A|) \le (t_1, t_2).$$

Since, in particular, code C is (t_1, t_2)-ST, by examining $\hat{Z}$, the decoder will conclude that X was the transmitted codeword. This is a contradiction.

Consider now the case $S = s_2$. By (3), we have $s = s_1$, $T = t_2$, $t = t_1$, $\rho = t_2 + s_2$, and $\tau = t_1 + s_1$. Therefore, (8) becomes

$$|D| = N(X, Y) \le \min\{t_1 + s_1, t_2\} \text{ and } |E| = N(Y, X) \le t_1 + t_2 + s_2. \tag{13}$$

We see that (13) is analogous to (10) with t_1 and t_2 and s_1 and s_2 reversed, leading to contradiction.

Therefore, assume that X and Y satisfy (9). Without loss of generality, let $\tau = t_1 + s_1$. Assume that the sequence

$$\hat{Z} = \hat{I}, \hat{E}, \hat{D}, \hat{I}, \ldots$$

is received. Since $S(Y; \hat{Z}) = (0, 0)$ (i.e., no skew), the decoder concludes that Y was the transmitted codeword. On the other hand, $S(X; \hat{Z}) = (N(X, Y), N(Y, X)) \le (t_1 + s_1, t_2 + s_2)$. Thus, the decoder either concludes that X was the transmitted codeword or it detects an error. This is a contradiction.

b) Assume now that

$$(t_1 - t_2)(s_1 - s_2) < 0. \tag{14}$$

Again we assume that the conditions are false. Namely, there exist X and Y with $N(X, Y) \le N(Y, X)$ satisfying (4), (5), (7), and

$$N(X, Y) \le t \text{ or } N(Y, X) \le \max\{\rho, t_1 + t_2 + s\}. \tag{15}$$

Combining (4), (5), (7), and (15), we conclude that X and Y satisfy either (9) or

$$N(X, Y) \le t \text{ and } N(Y, X) \le t_1 + t_2 + S, \tag{16}$$

or

$$N(X, Y) \leq \min\{\tau, T\} \text{ and } N(Y, X) \leq \max\{\rho, t_1 + t_2 + s\}. \quad (17)$$

We have already seen that (9) leads to a contradiction regardless of (3) or (14), so, it is enough to examine (16) and (17).

If $S = s_1$, (16) is contained in (10), while if $S = s_2$, (16) is contained in (13). Since (10) and (13) lead to contradiction, (16) cannot hold.

Assume that (17) holds. If $\max\{\rho, t_1 + t_2 + s\} = \rho$, then (17) is contained in (9), which leads to contradiction. Therefore, assume that $\max\{\rho, t_1 + t_2 + s\} = t_1 + t_2 + s$.

Consider first the case $S = s_1$. By (14), we have $s = s_2$, $T = t_2$, and $t = t_1$. Moreover, assume that $\tau = t_1 + s_1$. Thus, (17), becomes exactly (13), which leads to a contradiction. If $\tau = t_2 + s_2$, in particular, $t_2 \leq t_1 + s_1$, so, (17) is contained in (13), giving a contradiction.

Finally, consider the case $S = s_2$. By (14), we have $s = s_1$, $T = t_1$, and $t = t_2$. If $\tau = t_2 + s_2$, (17) becomes exactly (10), which leads to contradiction. If $\tau = t_1 + s_1$, in particular, $t_1 \leq t_2 + s_2$, so, (17) is contained in (10), giving a final contradiction. This completes the proof. □

B. The Decoding Algorithm

In this subsection we present an efficient skew correcting/detecting algorithm. We then prove that if C is a code meeting the conditions of Theorem 2.1 and X any codeword in C, whenever X is transmitted followed by other codewords, say Y and W, such that no transition from W arrives before reception of X is completed, and $\hat{Z}$ is the received sequence, then, if $S(X; \hat{Z}) \leq (t_1, t_2)$, the algorithm correctly concludes that X was the transmitted codeword, while if $(t_1, t_2) < S(X; \hat{Z}) \leq (t_1 + s_1, t_2 + s_2)$, then the algorithm will detect this situation when there are $t_2 + 1$ repeated arrivals. Namely, with this approach we prove that both the algorithm is correct and that the conditions of Theorem 2.1 are also sufficient.

ALGORITHM 2.1. (DECODING ALGORITHM). *Let the received sequence be* $\hat{Z} = x_1, x_2, \ldots, x_j, \ldots$. *Then*

Set the initial conditions $j \leftarrow 0$, $X \leftarrow \emptyset$, $B \leftarrow \emptyset$, and $W_l \leftarrow \emptyset$ for $1 \leq l \leq t_1 + t_2$.

```
START:  Set j ← j + 1.
If xj ∈ X, then:
 If |B| = t2 or xj ∈ B, then declare an uncor-
   rectable error and stop.
 Else, set B ← B ∪ {xj} and go to START.
Else, set X ← X ∪ {xj}, Wl ← Wl+1 for 1 ≤ l ≤ t1
   + t2 − 1 and W_{t1+t2} ← {xj}.
For each A ⊆ W_{t2−i+1} ∪ W_{t2−i+2} ∪ … ∪ W_{t1+t2−1} of size
   |A| = i,
 and for each 0 ≤ i ≤ t2, do:
If X − A ∈ C for some A, then output X − A.
 If A ∪ B ∈ C, then output A ∪ B, set X ← ∅,
   Wl ← ∅ for 1 ≤ l ≤ t1 + t2,
   B ← ∅ and go to START.
 Else, set X ← A ∪ B, Wl ← ∅ for 1 ≤ l ≤ t1 +
   t2, B ← ∅ and go to START.
Else, if X − A ∉ C for any A, go to START.
```

Notice that Algorithm 2.1 performs at most

$$\sum_{i=0}^{t_2} \binom{t_1 + i - 1}{i}$$

checks for each received transition. This is a fixed number depending only on t_1 and t_2, so the algorithm has low complexity. Basically, each check can be accomplished by verifying the membership of a vector in a code, which is easy to implement (using the encoding algorithm).

The set B stores the repeated arrivals. If the number of repeated arrivals exceeds t_2, then the skew between the transmitted codeword X and the received sequence $\hat{Z}$ exceeds (t_1, t_2), and the algorithm declares an uncorrectable error. It also declares an uncorrectable error when there is a second repeated arrival (i.e., $x_j \in B$). This means that transition x_j belongs to a codeword that is not the codeword transmitted after X. This algorithm does not handle skew between nonadjacent codewords, so when a situation like this is detected, an uncorrectable error is declared.

Before proving that the conditions in Theorem 2.1 are also sufficient, we give an example to familiarize the reader with Algorithm 2.1.

TABLE I
DECODING OF SEQUENCE IN EXAMPLE 2.1

j	x_j	B	X	W_1	W_2	A	$X - A$	Output
0		$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$			
1	4	$\emptyset$	$\{4\}$	$\emptyset$	$\{4\}$	$\emptyset$	$\{4\}$	
2	3	$\emptyset$	$\{3,4\}$	$\{4\}$	$\{3\}$	$\emptyset$ $\{4\}$	$\{3,4\}$ $\{3\}$	
3	1	$\emptyset$	$\{1,3,4\}$	$\{3\}$	$\{1\}$	$\emptyset$ $\{3\}$	$\{1,3,4\}$ $\{1,4\}$	
4	5	$\emptyset$ $\emptyset$	$\{1,3,4,5\}$ $\{1\}$	$\{1\}$ $\emptyset$	$\{5\}$ $\emptyset$	$\emptyset$ $\{1\}$	$\{1,3,4,5\}$ $\{3,4,5\}$	00111000
5	2	$\emptyset$ $\emptyset$	$\{1,2\}$ $\emptyset$	$\emptyset$ $\emptyset$	$\{2\}$ $\emptyset$	$\emptyset$	$\{1,2\}$	11000000
6	4	$\emptyset$	$\{4\}$	$\emptyset$	$\{4\}$	$\emptyset$	$\{4\}$	
7	5	$\emptyset$	$\{4,5\}$	$\{4\}$	$\{5\}$	$\emptyset$ $\{4\}$	$\{4,5\}$ $\{5\}$	
8	7	$\emptyset$	$\{4,5,7\}$	$\{5\}$	$\{7\}$	$\emptyset$ $\{5\}$	$\{4,5,7\}$ $\{4,7\}$	
9	2	$\emptyset$	$\{2,4,5,7\}$	$\{7\}$	$\{2\}$	$\emptyset$ $\{7\}$	$\{2,4,5,7\}$ $\{2,4,5\}$	
10	1	$\emptyset$	$\{1,2,4,5,7\}$	$\{2\}$	$\{1\}$	$\emptyset$ $\{2\}$	$\{1,2,4,5,7\}$ $\{1,4,5,7\}$	
11	2	$\{2\}$						
12	6	$\{2\}$	$\{1,2,4,5,6,7\}$	$\{1\}$	$\{6\}$	$\emptyset$ $\{1\}$	$\{1,2,4,5,6,7\}$ $\{2,4,5,6,7\}$	
13	8	$\{2\}$	$\{1,2,4,5,6,7,8\}$	$\{6\}$	$\{8\}$	$\emptyset$ $\{6\}$	$\{1,2,4,5,6,7,8\}$ $\{1,2,4,5,7,8\}$	
14	4	$\{2\}$						HALT!
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

EXAMPLE 2.1. Let $C = \{U, V, W\}$, where

$$U = 11000000 \leftrightarrow \{1, 2\}$$
$$V = 00111000 \leftrightarrow \{3, 4, 5\}$$
$$W = 01011111 \leftrightarrow \{2, 4, 5, 6, 7, 8\}.$$

Notice that $N(U, V) = 2$ and $N(V, U) = 3$, $N(U, W) = 1$ and $N(W, U) = 5$, and $N(V, W) = 1$ and $N(W, V) = 4$. As will be shown in Theorem 2.2, code C is (1, 1)-ST (2, 2)-SD.

Assume that V is transmitted followed by U, W, and other codewords, and the receiver obtains the sequence

$$\hat{Z} = \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & \cdots \\ 4 & 3 & 1 & 5 & 2 & 4 & 5 & 7 & 2 & 1 & 2 & 6 & 8 & 4 & \ldots \end{matrix}$$

Table I presents the execution of Algorithm 2.1 for the sequence $\hat{Z}$, with the relevant parameters at each step.

By looking at Table I and the received sequence $\hat{Z}$, we see that codeword V was transmitted first. Notice that $S(V; \hat{Z}) = (1, 1)$, so the algorithm correctly decodes V after the fourth arrival. Then U is decoded after the fifth arrival. The process starts again, and there is a new received sequence $\hat{Z} = 4, 5, 7, 2, 1, 2, 6, 8, 4, \ldots$. Now, $S(W; \hat{Z}) = (2, 2)$. This skew cannot be corrected, but it is detected when the algorithm sees two repeated arrivals.

C. The Sufficient Condition

We are ready to show that the necessary conditions in Theorem 2.1 are also sufficient.

THEOREM 2.2. *Let t_1, t_2, s_1, and s_2 be nonnegative integers and T, t, S, s, τ, ρ be defined as in Theorem* 2.1.

a) Assume that $(t_1 - t_2)(s_1 - s_2) \geq 0$. Let C be a code such that, for any $X, Y \in C$ with $N(X, Y) \leq N(Y, X)$, at least one of the following three conditions holds:

1) $N(X, Y) \geq \tau + 1$.
2) $N(X, Y) \geq T + 1$ *and* $N(Y, X) \geq \rho + 1$.
3) $N(X, Y) \geq 1$ *and* $N(Y, X) \geq t_1 + t_2 + S + 1$.

Then, code C is (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD.

b) Assume that $(t_1 - t_2)(s_1 - s_2) < 0$. Let C be a code such that, for any $X, Y \in C$ with $N(X, Y) \leq N(Y, X)$, at least one of the following four conditions holds:

1) $N(X, Y) \geq \tau + 1$.
2) $N(X, Y) \geq T + 1$ *and* $N(Y, X) \geq \rho + 1$.
3) $N(X, Y) \geq t + 1$ *and* $N(Y, X) \geq \max\{\rho + 1, t_1 + t_2 + s + 1\}$.
4) $N(X, Y) \geq 1$ *and* $N(Y, X) \geq t_1 + t_2 + S + 1$.

Then, code C is (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD.

PROOF. We prove parts a) and b) of the theorem simultaneously. The idea in the proof is to assume that if the algorithm accepts a vector which is in the code but not the correct one then it leads to a contradiction in the conditions in the theorem. It is enough to consider this case because in all other cases either the algorithm accepts the correct codeword or it will eventually detect an error.

Assume that Y is transmitted followed by other codewords, say V, W, ..., but only transitions from V may arrive before reception of Y is completed, giving a received sequence $\hat{Z}$. We have to prove that, if $S(Y; \hat{Z}) \leq (t_1, t_2)$, then the algorithm will correctly decode Y. Also, we have to show that the algorithm never produces a codeword different from Y if $S(Y; \hat{Z}) \leq (t_1 + s_1, t_2 + s_2)$.

By looking at Algorithm 2.1, we see that repeated arrivals are stored in the set B, and they do not influence the main part of the algorithm (after too many repeated arrivals, uncorrectable skew is detected). So, without loss of generality, we will assume that no repeated arrivals have occurred in $\hat{Z}$.

Let $\hat{Z}_l$ be the received sequence up to the arrival of the lth transition and let Z_l be the set of elements corresponding to the sequence $\hat{Z}_l$.

Let $m = m(Y; \hat{Z})$ and $r = r(Y; \hat{Z})$, where $m(Y; \hat{Z})$ and $r(Y; \hat{Z})$ are given by (1) and (2).

If $S(Y; \hat{Z}) \leq (t_1, t_2)$, there is an $A \subseteq \{x_{r-t_1-i+1}, x_{r-t_1-i+2}, \ldots, x_{r-1}\}$, $|A| = i \leq t_2$, such that $Z_r - A = Y$. So, the algorithm will correctly decode Y when transition r (i.e., the last transition in Y) arrives.

In order to prove the theorem, we need to show that the decoding algorithm never produces a codeword different from Y, i.e., for any $A \subseteq \{x_{l-t_1-i+1}, x_{l-t_1-i+2}, \ldots, x_{l-1}\}$ such that $Z_l - A \neq Y$, where $0 \leq |A| = i \leq t_2$, we prove that $Z_l - A \notin C$.

The proof is by contradiction. We assume that there exist an A and an l such that $F = Z_l - A \in C$. The false codeword F is different from the codeword that was sent, namely Y. The idea in the proof is to calculate bounds on $N(Y, F)$ and $N(F, Y)$, given the algorithm, and show that those bounds contradict the conditions in the theorem.

We first prove a general bound on $N(Y, F)$. Notice that $S(Y; \hat{Z}) \leq (t_1 + s_1, t_2 + s_2)$, $|Y - Z_m| \leq t_1 + s_1$ and that $Z_m \subset Y$. Hence,

$$N(Y, F) \leq |Y - Z_m| + |A| \leq (t_1 + s_1) + t_2 = t_1 + t_2 + s_1. \quad (18)$$

Next we prove a couple of bounds while considering the set $A \cap Z_m$.

- First, assume that $A \cap Z_m \neq \emptyset$.

  Let $A = A_1 \cup A_2$, where $A_1 \subseteq Z_m$ and $A_2 \subseteq Z_l - Z_m$. Let $|A_1| = i_1$ and $|A_2| = i_2$, therefore, $i = i_1 + i_2$. Since $A_1 = Z_m \cap A$, we have,

$$A_1 \subseteq \left\{x_{l-t_1-i+1}, x_{l-t_1-i+2}, \ldots, x_m\right\},$$

  therefore,

$$i_1 = |A_1| \leq \left|\left\{x_{l-t_1-i+1}, x_{l-t_1-i+2}, \ldots, x_m\right\}\right| = m - l + t_1 + i. \quad (19)$$

  Since $i_1 + i_2 = i$, (19) gives

$$l - m \leq t_1 + i_2. \quad (20)$$

  By (20), we have

$$N(F, Y) \leq l - m - |A_2| \leq (t_1 + i_2) - i_2 \leq t_1. \quad (21)$$

- On the other hand, if $A \cap Z_m = \emptyset$, we have that

$$N(Y, F) \leq |Y - Z_m| \leq t_1 + s_1. \quad (22)$$

Now we consider the value of l with respect to r. We have two cases: $l \leq r$ and $l > r$.

- Consider first the case $l \leq r$.

  If $l \leq m$, then $F \subset Y$, namely there are two codewords that are not unordered. Clearly, this leads to a contradiction to the hypothesis in the theorem. Therefore, $m + 1 \leq l \leq r$. Since $S(Y; \hat{Z}) \leq (t_1 + s_1, t_2 + s_2)$ and $l \leq r$, then, in particular,

$$N(F, Y) \leq t_2 + s_2. \tag{23}$$

  If $A \cap Z_m \neq \emptyset$, combining (18), (21), and (23), we obtain

$$N(F, Y) \leq \min\{t_1, t_2 + s_2\} \text{ and } N(Y, F) \leq t_1 + t_2 + s_1. \tag{24}$$

  It is easy to see that (24) contradicts conditions 1, 2, and 3 when a) holds and conditions 1, 2, 3, and 4 when b) holds. For instance, assume that a) holds and $T = t_1$, thus, $t = t_2$, $S = s_1$, $s = s_2$, $\rho = t_1 + s_1$, and $\tau = t_2 + s_2$. Condition 1 states that $\min\{N(F, Y), N(Y, F)\} \geq t_2 + s_2 + 1$, condition 2 states that $N(F, Y) \geq t_1 + 1$ and $N(Y, F) \geq t_1 + s_1 + 1$, while condition 3 states that $N(F, Y) \geq 1$ and $N(Y, F) \geq t_1 + t_2 + s_1 + 1$. In all cases, we contradict (24). Similarly, we obtain a contradiction when $T = t_2$, and also when b) holds.

  On the other hand, if $A \cap Z_m = \emptyset$, combining (23) and (22), we obtain

$$N(F, Y) \leq t_2 + s_2 \text{ and } N(Y, F) \leq t_1 + s_1. \tag{25}$$

  We can easily see that (25) contradicts the hypothesis.

  This shows that $l \leq r$ leads to contradiction.

- Now, assume that $l > r$, hence, $Y \subset Z_l$. Thus,

$$N(Y, F) \leq |A| \leq t_2. \tag{26}$$

  If $A \cap Z_m \neq \emptyset$, combining (21) and (26), we obtain

$$N(F, Y) \leq t_1 \text{ and } N(Y, F) \leq t_2. \tag{27}$$

  Clearly, (27) contradicts the hypothesis.

  If $A \cap Z_m = \emptyset$, since $S(Y; \hat{Z}) \leq (t_1 + s_1, t_2 + s_2)$, we have:

$$r - m = |Z_r - Y| + |Y - Z_m| \leq t_2 + s_2 + |Y - Z_m|. \tag{28}$$

  Let $A = A_1 \cup A_2$, where $A_1 \subseteq Y$ and $A_2 \subseteq Z_l - Y$. Let $i_1 = |A_1|$ and $i_2 = |A_2|$, thus, $i = i_1 + i_2$. Therefore,

$$N(F, Y) \leq (l - m) - i_2 - |Y - Z_m| = (l - r) + (r - m) - i_2 - |Y - Z_m|. \tag{29}$$

  Since

$$A_1 \subseteq A \cap Y \subseteq \left\{x_{l-t_1-i+1}, x_{l-t_1-i+2}, \ldots, x_r\right\},$$

  in particular,

$$i_1 = |A_1| \leq \left|\left\{x_{l-t_1-i+1}, x_{l-t_1-i+2}, \ldots, x_r\right\}\right| = r - l + t_1 + i, \tag{30}$$

  and since $i - i_1 = i_2$, (30) gives

$$l - r \leq t_1 + i_2. \tag{31}$$

  By (28) and (31), (29) becomes:

$$N(F, Y) \leq (t_1 + i_2) + (t_2 + s_2 + |Y - Z_m|) - i_2 - |Y - Z_m|$$
$$= t_1 + t_2 + s_2. \tag{32}$$

  Now combining (22), (26), and (32), we obtain

$$N(F, Y) \leq t_1 + t_2 + s_2 \text{ and } N(Y, F) \leq \min\{t_1 + s_1, t_2\}. \tag{33}$$

  Also (33) contradicts the hypothesis. □

The following corollary is immediate from Theorem 2.2.

COROLLARY 2.1. *Code C is (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD if and only if it is (t_2, t_1)-ST $(t_2 + s_2, t_1 + s_1)$-SD.*

Let us briefly examine special cases of the necessary and sufficient conditions given by Theorems 2.1 and 2.2. In particular, we will see that the conditions generalize known results [2]. Let us start with the case in which $\tau = t_1 + s_1 = t_2 + s_2$. Namely there is a symmetry in the maximum allowable skew.

THEOREM 2.3. *A code C is (t_1, t_2)-ST (τ, τ)-SD, if and only if, for any pair of codewords $X, Y \in C$ with $N(X, Y) \leq N(Y, X)$, at least one of the following three conditions occurs:*

1) $N(X, Y) \geq T + 1$ and $N(Y, X) \geq \tau + 1$.
2) $N(X, Y) \geq t + 1$ and $N(Y, X) \geq t + \tau + 1$.
3) $N(X, Y) \geq 1$ and $N(Y, X) \geq T + \tau + 1$.

Next consider the case $t_1 = t_2 = t$ and $s_1 = s_2 = s$.

THEOREM 2.4. *A code C is (t, t)-ST $(t + s, t + s)$-SD, if and only if, for any pair of codewords $X, Y \in C$ with $N(X, Y) \leq N(Y, X)$, at least one of the following two conditions occurs:*

1) $N(X, Y) \geq t + 1$ and $N(Y, X) \geq t + s + 1$.
2) $N(X, Y) \geq 1$ and $N(Y, X) \geq 2t + s + 1$.

Using Theorem 2.4, we conclude that the codes in Examples 1.2 and 2.1 are (1, 1)-ST (2, 2)-SD.

Next, consider the case in which $s_1 = s_2 = 0$, i.e., the necessary and sufficient conditions for a code to be (t_1, t_2)-ST.

THEOREM 2.5. *Let $t = \min\{t_1, t_2\}$. A code C is (t_1, t_2)-ST if and only if, for any pair of codewords $X, Y \in C$ with $N(X, Y) \leq N(Y, X)$, at least one of the following two conditions occurs:*

1) $N(X, Y) \geq t + 1$.
2) $N(X, Y) \geq 1$ *and* $N(Y, X) \geq t_1 + t_2 + 1$.

Finally, we make $t_1 = t_2 = 0$ in order to obtain necessary and sufficient conditions for (s_1, s_2)-SD codes. The result is given in the next theorem:

THEOREM 2.6. *Let $s = \min\{s_1, s_2\}$ and $S = \max\{s_1, s_2\}$. A code C is (s_1, s_2)-SD if and only if, for any pair of codewords $X, Y \in C$ with $N(X, Y) \leq N(Y, X)$, at least one of the following two conditions occurs:*

1) $N(X, Y) \geq s + 1$.
2) $N(X, Y) \geq 1$ *and* $N(Y, X) \geq S + 1$.

Theorems 2.5 and 2.6 were known [2].

III. CONSTRUCTION OF (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD CODES

In this section, we present codes satisfying the sufficient conditions given in Theorem 2.2. A solution to the problem is provided by the so-called error correcting/all unidirectional error detecting (EC/AUED) codes [5], [6], [7], [8]. In effect, a code is τ-EC/AUED if and only if, for any pair of distinct codewords X and Y, $N(X, Y) \geq \tau + 1$ [8]. Given t_1, t_2, s_1, and s_2, $\tau = \min\{t_1 + s_1, t_2 + s_2\}$, by Theorem 2.2, a τ-EC/AUED code is (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD. Efficient constructions of τ-EC/AUED codes can be found in [5], [6], [7].

However, in general, there are better constructions for (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD codes. These constructions involve some of the techniques given in [4]. We repeat them here without proof.

CONSTRUCTION 3.1. Consider the $s + 1$ vectors $\underline{w}_0, \underline{w}_1, \ldots, \underline{w}_s$ of length s defined as follows:

$$\underline{w}_i = \overbrace{00\ldots0}^{i}\overbrace{11\ldots1}^{s-i}.$$

Given any integer i and an integer $m > 0$, we denote by $\langle i \rangle^m$ the unique integer j, $0 \leq j \leq m - 1$, such that $i \equiv j (\mathrm{mod}\ m)$.

Consider the following matrix, denoted $B(w, s)$, with w rows $\underline{u}_0, \underline{u}_1, \ldots, \underline{u}_{w-1}$ and s columns: row $\underline{u}_i$ is given by vector $\underline{w}_j$, where $j = \langle i \rangle_{s+1}$.

For instance, if $w = 9$ and $s = 3$, we have

$$B(9,3) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Given k information bits, the next construction encodes them into a code C of length n.

CONSTRUCTION 3.2. Let $0 \leq a \leq b$ be integers. Choose an $[n', k, 2a + 2]$ error correcting (EC) code C' in which the Hamming weights are at least two apart (e.g., all the weights are even). Consider the matrix $B(\lceil (n' + 1)/2 \rceil, b - a - 1)$ as given by Construction 3.1. Let $\underline{u}$ be an information vector. Then proceed as follows:

1) Encode $\underline{u}$ into a vector $\underline{v} \in C'$.
2) Let j be the Hamming weight of $\underline{v}$. Then append to $\underline{v}$ row $\lfloor j/2 \rfloor$ of matrix $B(\lceil (n' + 1)/2 \rceil, b - a - 1)$.
3) Append the complement of the binary representation of $\lfloor j/(2a + 2) \rfloor$ if $a + 1 \geq b - a$ or the complement of the binary representation of $\lfloor j/(2b - 2a) \rfloor$ if $a + 1 < b - a$.

The next theorem was proven in [4].

THEOREM 3.1. *Let $X = (\underline{v}_1, \underline{r}_1, \underline{s}_1)$ and $Y = (\underline{v}_2, \underline{r}_2, \underline{s}_2)$ be codewords that are obtained by using Construction 3.2 with parameters a and b, $a \leq b$. Namely, $\underline{v}_1$ and $\underline{v}_2$ are even-weight codewords in an $[n', k, 2a + 2]$ code, $\underline{r}_1$ and $\underline{r}_2$ are the tails corresponding to rows in the $B(\lceil (n' + 1)/2 \rceil, b - a - 1)$ matrix, and $\underline{s}_1$ and $\underline{s}_2$ are the tails as described in the third step of the construction. Then at least one of the following two conditions occurs:*

1) $\min\{N(X, Y), N(Y, X)\} \geq a + 1$

or

2) $\min\{N(X, Y), N(Y, X)\} \geq 1$ *and* $\max\{N(X, Y), N(Y, X)\} \geq b + 1$.

From Theorems 2.2 and 3.1, the following corollary is clear:

COROLLARY 3.1. *Let t_1, t_2, s_1, and s_2 be nonnegative integers, and let τ and S be defined as in Theorem 2.1. Given k information bits, construct a code C using Construction 3.2 with $a = \tau$ and $b = t_1 + t_2 + S$. Then, C is (t_1, t_2)-ST $(t_1 + s_1, t_2 + s_2)$-SD.*

EXAMPLE 3.1. Assume that we want to construct a (1, 1)-SD (2, 2)-ST code with $k = 20$ information bits. In this case, we have $t_1 = t_2 = s_1 = s_2 = 1$, therefore $\tau = 2$ and $S = 1$. By the observation at the beginning of this section, a 2-EC/AUED code is (1, 1)-SD (2, 2)-ST. By using the constructions in [7], for instance, there is a 2-EC/AUED code with 21 information bits and 18 redundant bits.

If we use Construction 3.2 with $a = 2$ and $b = 4$, first we encode into a BCH code with minimum distance six; we need 11 bits to achieve this. Then, we add the second tail, that has length $b - a - 1 = 1$. The third tail unorders the code similarly to the Berger construction [1], [3], by writing the complement of the binary representation of the weight of the current codeword divided by the minimum distance six; we need an extra three bits to achieve this. Therefore, the total redundancy is 15 bits.

IV. CONCLUSIONS

We have devised a novel scheme based on coding techniques that allows delay-insensitive communication on parallel channels. We gave a precise mathematical definition of the concept of skew and proved necessary and sufficient conditions for codes that can tolerate a predetermined amount of skew and detect a higher amount of skew when this predetermined amount is exceeded. We have constructed codes satisfying the necessary and sufficient conditions and devised efficient encoding and decoding algorithms.

ACKNOWLEDGMENT

We are grateful to one of the referees whose comments helped in improving the precision of the presentation.

REFERENCES

[1] J.M. Berger, "A note on error detecting codes for asymmetric channels," *Information and Control*, vol. 4, pp. 68-73, Mar. 1961.
[2] M. Blaum and J. Bruck, "Coding for skew-tolerant parallel asynchronous communications," *IEEE Trans. Information Theory*, vol. 39, pp. 379-388, Mar. 1993.
[3] M. Blaum and J. Bruck, "Unordered error-correcting codes and their

applications," *Proc. FTCS-22*, pp. 486-493, Boston, July 1992.

[4] M. Blaum, J. Bruck, and L.H. Khachatrian, "Construction of skew-tolerant and skew-detecting codes," *IEEE Trans. Information Theory*, vol. 39, pp. 1,751-1,757, Sept. 1993.

[5] M. Blaum and H. van Tilborg, "On t-error correcting/all unidirectional error detecting codes," *IEEE Trans. Computers*, vol. 38, pp. 1,493-1,501, Nov. 1989.

[6] F.J.H. Boinck and H. van Tilborg, "Constructions and bounds for systematic tEC/AUED codes," *IEEE Trans. Information Theory*, vol. 36, no. 6, pp. 1,381-1,390, Nov. 1990.

[7] J. Bruck and M. Blaum, "New techniques for constructing EC/AUED codes," *IEEE Trans. Computers*, vol. 41, no. 10, pp. 1,318-1,324, Oct. 1992.

[8] D.K. Pradhan, "A new class of error-correcting detecting codes for fault-tolerant computer application," *IEEE Trans. Computers*, vol. 29, pp. 471-481, June 1980.

[9] T. Verhoeff, "Delay-insensitive codes—an overview," *Distributed Computing*, vol. 3, pp. 1-8, 1988.

Mario Blaum received the degree of Licenciado from the University of Buenos Aires in 1977, the MSc degree from the Israel Institute of Technology in 1981, and the PhD degree from the California Institute of Technology in 1984, all in mathematics. From January to June 1985 he was a research fellow at the Department of Electrical Engineering at Caltech. In August 1985 he jointed the IBM Research Division at the Almaden Research Center, where he is currently a research staff member. From September 1990 to September 1991 he was a consulting professor at Stanford University teaching a course in error-correcting codes.

Dr. Blaum's research interests include error-correcting codes, storage technology, combinatorics, and neural networks. He is a senior member of the IEEE.

Jehoshua Bruck received the BSc and MSc degrees in electrical engineering from the Technion, Israel Institute of Technology, in 1982 and 1985, respectively. He received the PhD degree in electrical engineering from Stanford University in 1989.

Dr. Bruck is currently an associate professor of computation and neural systems and electrical engineering at the California Institute of Technology. He has extensive industrial experience, including serving as manager of the Foundations of Massively Parallel Computing Group at the IBM Almaden Research Center from 1990 to 1994, a research staff member at the IBM Almaden Research Center from 1989 to 1990, and a researcher at the IBM Haifa Science Center from 1982 to 1985. Dr. Bruck is the recipient of a 1992 Outstanding Innovation Award for his work on "Harmonic Analysis of Neural Networks," a 1994 National Science Foundation Young Investigator award, a 1995 Alfred P. Sloan Research Fellowship, and a 1994 IBM Outstanding Technical Achievement Award for his contributions to the design and implementation of the SP-1, the first IBM scalable parallel computer. He is also the recipient of five IBM Plateau Invention Achievement Awards and holds 16 patents.

Dr. Bruck's research interests include parallel and distributed computing, fault-tolerant computing, error-correcting codes, and neural networks.